

The Claims

Claims 1-27. (Canceled).

28. (Currently amended) A computer-readable medium having stored thereon a data structure, the data structure comprising:

a presentation time portion indicating when audio data is to be rendered;

a data portion that can include audio data or a pointer to a chain of additional data structures that include the audio data; and

a flag portion indicating to a kernel-mode transform filter whether the data portion includes the pointer to the chain of additional data structures, wherein the flag portion further includes an event incomplete flag that can be set to indicate that data identified in the data portion extends beyond a buffer pointed to by a pointer maintained in the data portion.

29. (Original) A computer-readable medium as recited in claim 28, wherein the data structure further comprises a structure byte count portion that identifies the size of the data structure.

30. (Original) A computer-readable medium as recited in claim 28, wherein the data structure further comprises an event byte count portion that identifies a number of data bytes that are referred to in the data portion.

31. (Original) A computer-readable medium as recited in claim 28, wherein the data structure further comprises a channel group portion that identifies which of a plurality of channel groups the data identified in the data portion corresponds to.

32. (Canceled).

33. (Original) A computer-readable medium as recited in claim 28, wherein the data structure further comprises a byte position portion including an identifier of where the data structure is situated among a plurality of data structures received from an application.

34. (Original) A computer-readable medium as recited in claim 28, wherein the data structure further comprises a next event portion including an identifier of a next data structure in a chain of data structures.

35. (Original) A computer-readable medium as recited in claim 28, wherein:

the data portion can further include a pointer to a data buffer; and

the flag portion indicates whether the data portion includes either the pointer to the chain of additional data structures or one of either the audio data or the pointer to the data buffer.

36. (Original) A computer-readable medium as recited in claim 35, further comprising an event byte count portion that identifies, if the data portion does not include the pointer to the chain of additional data structures, whether the data portion includes the audio data or a pointer to the data buffer.

Claims 37-43. (Canceled).

44. (Currently amended) A system comprising:

a first module implemented in kernel-mode and coupled to receive audio data from hardware, wherein the first module is to process the audio data by obtaining a data packet structure into which the audio data can be placed, wherein the data packet structure includes:

a data portion that can include one of: the audio data, a pointer to a chain of additional data packet structures that include the audio data, and a pointer to a data buffer, and

an event byte count portion that identifies, if the data portion does not include the pointer to the chain of additional data packet structures, whether the data portion includes the audio data or a pointer to the data buffer;

a second module implemented in kernel mode and coupled to communicate processed audio data to an application executing in user-mode; and

a third module, implemented in kernel-mode, to receive the audio data ~~from~~ from the first module, process the audio data, and communicate the processed audio data to the second module.

45. (Original) A system as recited in claim 44, wherein the first module is further to process the audio data before forwarding the audio data to the second module.

46. (Canceled).

47. (Original) A system as recited in claim 44, further comprising additional modules, situated between the first and third modules, to further process the audio data.

48. (Currently amended) One or more computer-readable media having stored thereon a series of instructions that, when executed by one or more processors of a computer, causes the one or more processors to perform acts including:

maintaining a pool of memory available for allocation to a plurality of transform filters executing at a privileged level;

allocating a portion of the pool of memory to one of the plurality of transform filters to use to store audio data, wherein the portion comprises sufficient memory to store a data structure including:

a data portion that can include one of: audio data, a pointer to a chain of additional data structures that include the audio data, and a pointer to a data buffer;

a structure byte count portion that identifies the size of the data structure;

a channel group portion that identifies which of a plurality of channel groups the data identified in the data portion corresponds to;

a presentation time portion indicating when audio data is to be rendered;

a flag portion indicating whether the data portion includes either the pointer to the chain of additional data structures or one of either the audio data or the pointer to the data buffer; and

an event byte count portion that identifies, if the data portion does not include the pointer to the chain of additional data structures, whether the data portion includes the audio data or a pointer to the data buffer; and

returning the allocated portion to the pool of memory after the plurality of transform filters have finished processing the audio data.

49. (Original) One or more computer-readable media as recited in claim 48, wherein the privileged level comprises kernel-mode.

50. (Canceled).

51. (Original) One or more computer-readable media as recited in claim 48, wherein the portion comprises a data buffer to store a plurality of audio data messages.

52. (Original) One or more computer-readable media as recited in claim 48, wherein the series of instructions, when executed, further cause the one or more processors to perform acts including requesting additional memory, from a memory manager, to add to the pool of memory.

53. (Original) One or more computer-readable media as recited in claim 52, wherein the series of instructions, when executed, further cause the one or more processors to perform acts including requesting additional non-paged memory from the memory manager to add to the pool of memory.

Claims 54-58. (Canceled).

59. (Currently amended) One or more computer-readable media having stored thereon a transform filter for execution in kernel-mode that, when executed in kernel-mode by one or more processors of a computer, causes the one or more processors to implement:

a ConnectOutput interface to allow identification to the transform filter of a next transform filter in a transform filter graph to which audio data packets should be communicated by the transform filter; and

a PutMessage interface to allow the audio data packets to be communicated to the next transform filter;

wherein each of the audio data packets includes a flag portion that includes an event incomplete flag that can be set to indicate that data identified in a data

portion of the audio data packet extends beyond a buffer pointed to by a pointer maintained in the data portion.

60. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a SetState interface to allow a state of the transform filter to be set, including a run state and a stop state.

61. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a DisconnectOutput interface to allow a previously identified next transform filter to be changed.

62. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a SetParameters interface to allow operational parameters of the transform filter to be set.

63. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a GetParameters filter to allow operational parameters previously sent to the transform filter to be retrieved.

64. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a GetMessage interface to allow other transform filters in the transform filter graph to obtain data structures for the audio data packets.

65. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a GetBufferSize interface to allow other transform filters in the transform filter graph to obtain a size of data buffers allocated by the transform filter.

66. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a GetBuffer interface to allow other transform filters to obtain data buffers for storage of audio data corresponding to an audio data packet.

67. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter further causes the one or more processors to implement a PutBuffer interface to allow other transform filters to return data buffers to a memory pool for re-allocation.

68. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter comprises a sequencer filter to reorder the audio data packets by timestamp and to delay forwarding the audio packets to the next transform filter until an appropriate time.

69. (Original) One or more computer-readable media as recited in claim 59, wherein the transform filter comprises an allocator filter to obtain memory from a memory manager and make portions of the obtained memory available to other transform filters.